

Remarks

This Amendment responds to the Office Action (“the Action”) mailed March 21, 2007. Reconsideration of the application is respectfully requested in view of the foregoing amendments and following remarks. Reconsideration of the application is respectfully requested in view of the foregoing amendments and following remarks. Claims 1-17, 19, and 20 are pending in the application. Claim 18 has been canceled. No claims have been allowed. Claims 1, 6, 16, and 20 are independent.

Cited Art

U.S. Patent No. 7,043,490 to Choy, et al. (“Choy”) is titled “Method, System, and Program Product to Support Multiple Content-Management Data Models”

“Java Look and Feel Design Guidelines: Advanced Topics” is by Sun Microsystems, Inc. (“Java”).

“Supporting Document and Data Views of Source Code” is by Collard, et al. (“Collard”).

Amendments

Editorial amendments have been made to claims 1, 16, and 17. No new matter is added.

Rejection of Claims under 35 U.S.C. § 101

The Action rejects claims 1-5 16, 17, and 19 under 35 U.S.C. § 101, alleging that the claims are not directed to statutory subject matter. Specifically, the Action alleges that these claims fail to recite “a practical application.” [Action, at §4, page 3.]

The claims have since been amended. Claim 1, as amended, recites that “the source code viewer can create a version of source code using a selected segment of source code.” Claim 16, as amended, recites “wherein the code layer viewer edits the source code file to create a version of the source code file that includes an indicated code segment choice.” As such, both claims 1 and 16, and therefore claims 2-5, 17 and 19 which respectively depend from them, recite the result of “creat[ing] a version of [] source code.” The claims therefore recite a practical

application and are therefore directed to statutory subject matter. Applicants therefore request that the rejection of claims 1-5 and 16, 17, and 19 under 35 U.S.C. § 101 be withdrawn.

Rejection of Claims Under 35 U.S.C. § 102 over Choy

The Action rejects claims 1-7, 9, 10, and 16, 17, 19, and 20 under 35 U.S.C. § 102 as being anticipated by Choy. Applicants respectfully disagree and traverse the rejection. For a 102 rejection to be proper, the cited art must show each and every element as set forth in a claim. (See MPEP § 2131.01.) However, the cited art does not describe each and every element. Accordingly, applicants request that the rejection be withdrawn. Claims 1, 6, 16, and 20 are independent.

Independent Claim 6

Claim 6 recites, in part:

receiving a file comprising source code with code layer choice identifiers;
displaying the source code on a computer terminal comprising a code
layer choice;
receiving an indication of a code layer choice; and
creating a version of source code comprising the indicated code layer
choice.

[Emphasis added.] For example, the Application describes an example of a source code file including layer identifiers at Figure 1 and accompanying text:

Figure 1 is a diagram representing a source code file 100 with exemplary code layer identifiers defining code segment boundaries. . . .

As shown, a source code file includes an identifier 102 that identifies the beginning of a first code segment 116. Additionally, the source code file includes an identifier that identifies the beginning of a second code segment 104. A source code programmer signals source code segment choices by placing the identifiers in the source code. The intent of placing the identifiers is to signal to a code viewer to offer a source code choice. In this example, the second identifier 104 also identifies the end of the first code segment 116 and a third identifier 106, identifies the end of the second code segment 118.

In this source code file example 100, a source code file includes two code layer choices (e.g., choice 1 or 2, and choice 3 or 4). The code segments selectable by a user are defined using the identifiers. Namely, identifiers a, b, and c, 102-106, define the first code layer choice 116 or 118, and identifiers a, b, and c, 108-112, define the second code layer choice 122 or 124.

[Application, at page 4, line 14 to page 5, line 19; emphasis added.] The Application also describes exemplary using source code and creating versions of source code at page 8:

Figure 3 is an exemplary diagram of a process 300 using a code layer viewer to create a source code version.

In this example, a code layer viewer 302 receives source code as input 304. The source code includes identifiers of at least one code layer choice. . . .

In this example, the user is presented with two code segments 308, 310, and the user makes a code layer choice by selecting either the first 308 or second 310 code segment.

. . . .
[A] user selects one of the code segments to be used in a source code version 312. . . . After evaluating the performance of or debugging a version of the source code 312, a user may decide to create another version of the program for testing or debugging purposes. For this reason, and others, a copy of the source code 316 might be made to iteratively create other versions for test.

[Application, at page 8, lines 1-26.]

Choy, which is directed toward managing multimedia content and which does not discuss the use or manipulation of source code, does not describe either “receiving a file comprising source code with code layer choice identifiers” or “creating a version of source code comprising the indicated code layer choice” as recited in claim 6. Choy is directed toward “[m]ethods, systems, and program products for managing multimedia content.” [Choy, at Abstract.] Toward this end, Choy provides “an extensible content management system such that a user can introduce appropriate metadata and schema and new APIs to support new high-level data models.” [Choy, at column 3, lines 43-46.]

In its rejection of claim 6, Choy cites to column 5, lines 52-65 and 15-24 for the language of the claim quoted above. These portions of Choy, however, are related to the relations of items with each other in a Relational Database Management System. [Choy, at Figure 2 and column 4, lines 23-24.] At lines 52-65, Choy discusses the relations of components, with each other using Links. [Choy, at column 5, lines 52-55.] Thus, this passage of Choy talks only of creating specific types of relationships between items in Choy’s database. The other cited passage of Choy, at lines 15-24, talks about versions of item types. Specifically, it discusses keeping track of “versions” of items by using version numbers. [Choy, at column 5, lines 18-19.]

Neither passage, however, discussed source code. In fact, nowhere in Choy is the term “source code” ever used. Choy does not even discuss “code” except in the context of claiming

its invention in the form of computer-readable code (*see, e.g.*, claims 25 and 31) and an Access Control List code, which is not source code. [Choy, at column 5, lines 33-34.] Thus, Choy does not describe the use or manipulation of source code and therefore does not describe the above-quoted language of claim 6.

Choy's system, which relies on a database system to mask distribution of data over various storage devices, does not describe "receiving a file comprising source code with code layer choice identifiers" as is recited in claim 6. Choy relies on a multi-level system which allows representations of content at high levels irrespective of how data constituting the content is mapped to data storage. [Choy, at Abstract.] Thus, as Choy demonstrates, a relational database is used to mask the differences in high-level data representation and low-level data storage:

To accommodate more than one content data model in a high-level data model, we provide an extensible content management system such that a user can introduce appropriate metadata and schema and new APIs to support new high-level data models. *This extensibility is enabled by an architecture that is built around a low-level infrastructure supporting a physical data model and implemented through a hierarchical structure of RDBMS tables of content and content metadata*, as items, components, sub-components, and attributes, with corresponding rows and columns (attributes) within the rows. The architecture and an associated infrastructure and physical data model are used as building blocks to support high-level data models. *This physical data model is independent of application-specific semantics, maps directly and efficiently to a relational database designed to exploit RDBMS capabilities, and can be enriched over time to expand its functions and/or to exploit new RDBMS capabilities.*

[Choy, at column 3, lines 42-59; emphasis added; *see also* Figure 2.] Toward this, as Choy demonstrates at Figure 6, the system of Choy relies on resources which can be arbitrarily split or broken up:

As illustrated in FIG. 6, a Resource Reference, 59, contains a reference to a Resource, 61, managed by a Resource Manager of the content management system. A Resource Manager is a special-purpose server that manages resources of a particular type. Examples are an Object Server, which manages a large volume of binary objects, and a Stream Server, which manages and delivers streaming objects (e.g., audio and video content). The extensible content management system in our design is able to accommodate new Resource Managers, each of which is specially designed to manage a specific type of resources, with its specific representation and methods that are not necessarily modeled by the physical data model described here.

[Choy, at column 6, lines 38-53.] Thus, while Choy relies on the ability to keep different resources on different storage devices and in different files, it does not appear to describe “receiving a file comprising source code with code layer choice identifiers” as is recited in claim 6. Furthermore, the cited passage of Choy at column 5, lines 52-65, which describes links between items in a relational database, does not describe the usage of a source code file with code layer identifiers. Thus, Choy also does not describe the above-quoted language of claim 6.

Therefore, for at least these reasons, Choy does not appear to describe each and every element as set forth in claim 6. The rejection of claim 6 over Choy is thus improper. Applicants respectfully note that claim 6 is allowable and request that claim 6, as well as dependent claims 7, 9, and 10, be allowed.

Independent Claim 1

Claim 1, as amended, recites, in part:

wherein the identifiers indicate to a source code viewer that a choice is displayed to select the first or second segments such that the source code viewer creates a version of source code using a selected segment of source code

The Action rejects claim 1 over a single passage also cited in the rejection of claim 6.

[Action, at page 3.] Thus, for at least the reasons cited above with respect to claim 6, Choy does not appear to describe each and every element as set forth in claim 1. The rejection of claim 1 over Choy is thus improper. Applicants respectfully note that claim 1 is allowable and request that claim 1, as well as dependent claims 2-5, be allowed.

Independent Claim 16

Claim 1, as amended, recites:

a memory comprising a code layer viewer component, a source code file comprising source code and code layer identifiers;

...
wherein the code layer viewer edits the source code file to create a version of the source code file that includes an indicated code segment choice.

The Action rejects claim 16 “as per the rejection for claim 1.” [Action, at page 4.] Thus, for at least the reasons cited above with respect to claims 1 and 6, Choy does not appear to describe each and every element as set forth in claim 16. The rejection of claim 16 over Choy is

thus improper. Applicants respectfully note that claim 16 is allowable and request that claim 16, as well as dependent claim 17, be allowed.

Independent Claim 20

Claim 20 recites, in part:

receiving a media content file comprising common content and plural media content alternatives

The Action rejects claim 20 over column 2, lines 41-52 of Choy [Action, at page 4.] The cited passage describes “a method of representing a multimedia content management object in a relational database adapted for representing multimedia content management data in one or more relational database tables” as well as “managing the multimedia content database management system.” [Choy, at column 2, lines 41-46.] Applicants note that, while the cited passage of Choy describes “relational database tables” it also notes that “any columns of these tables may contain pointers to other items or to non-structured content resources.” [Choy, at column 2, lines 46-52.]

Thus, for at least the reasons cited above with respect to claim 6, Choy does not appear to describe each and every element as set forth in claim 20. The rejection of claim 20 over Choy is thus improper. Applicants respectfully note that claim 20 is allowable and request that claim 20 be allowed.

Rejections under 35 U.S.C. § 103(a)

The Action rejects claims 11-15 under 35 U.S.C. § 103(a) over the Choy in combination with Java. The Action also rejects claims 8 and 19 under 35 U.S.C. § 103(a) over the Choy in combination with Collard. To establish a prima facie case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. [See, MPEP § 2142.]

Each of claims 8, 11-15, and 19 depend from claims 6 or 16 and recite additional patentable language. In the interest of expediency, Applicants do not belabor the individual

language of each claim but note that, for at least the reasons given above, Choy fails to teach or suggest each and every element of these dependent claims. Applicants do not find further disclosure in Java or Collard.

For at least these reasons, the rejections of claims 8, 11-15, and 19 are improper and fail to establish prima facie obviousness according to the standard set forth in MPEP § 2142. Thus, Applicants respectfully note that the claims are allowable and respectfully request their allowance.

Request for Interview

If any issues remain, the Examiner is formally requested to contact the undersigned attorney prior to issuance of the next Office Action in order to arrange a telephonic interview. It is believed that a brief discussion of the merits of the present application may expedite prosecution. Applicants submit the foregoing formal Amendment so that the Examiner may fully evaluate Applicants' position, thereby enabling the interview to be more focused.

This request is being submitted under MPEP § 713.01, which indicates that an interview may be arranged in advance by a written request.

Conclusion

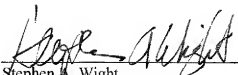
Claims 1-17, 19, and 20 are allowable. Applicants respectfully request allowance of the application.

Respectfully submitted,

KLARQUIST SPARKMAN, LLP

One World Trade Center, Suite 1600
121 S.W. Salmon Street
Portland, Oregon 97204
Telephone: (503) 595-5300
Facsimile: (503) 595-5301

By


Stephen K. Wight
Registration No. 37,759